Dynamic Programming to Coin Change Problem

syscl

Let the $d_1, \cdots, d_k$ be the denominations, n be the amount of money, c_p be the minimum number of coins of current denomination system needed to make change for p. Let

$$c_0 = 0 \tag{1}$$
$$c_p = \min_{1 \le i \le k} (1 + c_{p-d_i}) \tag{2}$$

then c_p is optimal solution to p.

Before we completes the proof, We have to prove the following lemmas.

Lemma 1 Let $X = \{x_1, \cdots, x_k\}$ be an optimal solution where x_i is the number of d_i being used. If $P \subseteq X$ and $\alpha = \sum_{x_i \in P} d_i x_i$, then P is an optimal solution to α.

Proof. If P is not an optimal solution to P, there there must exist a set $Q \subseteq X$ such that $\alpha = \sum_{x_i \in Q} d_i x_i$ and $\sum_{x \in Q} x < \sum_{x \in P} x$. If that is the case,

$$\sum_{x \in Q \cup (X-P)} x < \sum_{x \in X} x \tag{3}$$

and this is contradict to X is an optimal solution since $Q \cup (X - P)$ is a valid solution. □

Corollary $X - P$ is optimal solution to $n - \alpha$.

We now go to prove the recurrence relation c_p is optimal solution.

Proof. For the base case where $c_0 = 0$ is trivially correct. By **Lemma 1**, the optimal solution to the coin changing problem is composed of optimal solutions to smaller subproblems, which completes the proof. □